L Number	Hits	Search Text	DB	Time stamp
1	3521	((711/123) or (711/126) or (711/128) or (711/129) or (711/3) or (711/153) or (711/171) or (711/173) or (712/227) or (712/229) or (714/45) or (717/127) or (717/128)).CCLS.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:16
2	86	(((711/123) or (711/126) or (711/128) or (711/129) or (711/3) or (711/153) or (711/171) or (711/173) or (712/227) or (712/229) or (714/45) or (717/127) or (717/128)).CCLS.) and (cache adj allocat$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:17
3	0	(((711/123) or (711/126) or (711/128) or (711/129) or (711/3) or (711/153) or (711/171) or (711/173) or (712/227) or (712/229) or (714/45) or (717/127) or (717/128)).CCLS.) and (cache adj allocat$3) and (trace adj array)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:17
4	6	(((711/123) or (711/126) or (711/128) or (711/129) or (711/3) or (711/153) or (711/171) or (711/173) or (712/227) or (712/229) or (714/45) or (717/127) or (717/128)).CCLS.) and (cache adj allocat$3) and (trace or debug)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:39
5	85	((cache adj allocat$3) and (trace or debug)) not ((((711/123) or (711/126) or (711/128) or (711/129) or (711/3) or (711/153) or (711/171) or (711/173) or (712/227) or (712/229) or (714/45) or (717/127) or (717/128)).CCLS.) and (cache adj allocat$3) and (trace or debug))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:39
6	3	((cache adj allocat$3) with (trace or debug)) not ((((711/123) or (711/126) or (711/128) or (711/129) or (711/3) or (711/153) or (711/171) or (711/173) or (712/227) or (712/229) or (714/45) or (717/127) or (717/128)).CCLS.) and (cache adj allocat$3) and (trace or debug))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:42
7	1543	(system adj cache) and (trace array)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:42
8	1	(system adj cache) and (trace adj array)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:42
9	197	trace adj array	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:46
10	6	(trace adj array) with cache	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:43
11	191	(trace adj array) not ((trace adj array) with cache)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/11/20 15:46

L Number	Hits	Search Text	DB	Time stamp
1	0	(system adj on) adj2 chip	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 15:01
2	18	((system adj on) or (system-on)) adj2 chip	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 15:01

L Number	Hits	Search Text	DB	Time stamp
1	161	cache adj3 trace	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 13:26
2	33	cache adj trace	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 13:26

L Number	Hits	Search Text	DB	Time stamp
1	274	(717/124).CCLS.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 11:15
2	69	((717/124).CCLS.) and trace	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 11:16
3	20	((717/124).CCLS.) and (trace with memory)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 11:16

L Number	Hits	Search Text	DB	Time stamp
3	0	(trace adj array) with ((cache or memory) adj allocat$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 09:29
4	0	(trace adj array) with ((cache or memory) adj3 allocat$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 09:29
5	14	(trace adj3 (block or memory)) with ((cache or memory) adj3 allocat$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 09:54
7	1	(trace adj (block or memory)) with ((cache or memory) adj allocat$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 09:29
6	13	(trace adj3 (block or memory)) with ((cache or memory) adj allocat$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 09:30
8	1	((trace adj3 (block or memory)) with ((cache or memory) adj3 allocat$3)) not ((trace adj3 (block or memory)) with ((cache or memory) adj allocat$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/12/10 09:54

TDB-ACC-NO: NN9207138

DISCLOSURE TITLE: Trace Array.

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, July 1992, US

VOLUME NUMBER: 35

ISSUE NUMBER: 2

PAGE NUMBER: 138 - 140

PUBLICATION-DATE: July 1, 1992 (19920701)

CROSS REFERENCE: 0018-8689-35-2-138

DISCLOSURE TEXT:

- A logic analyzer function is implemented internal to a multi-chip module (MCM) and controllable by microcode. The analyzer stores signal data in an array and provides different trace modes, selectable triggers, selectable inputs, and an external (common) trigger line. The external trigger connects to all trace arrays in a multiprocessor environment and allows synchronizing all the logic traces when a trigger occurs.

- Enough signals are stored in the trace array to allow efficient debug of the system. Both MCM internal only and externally available signals are stored in the trace array. Signals being stored include major control interfaces and sequencers.

- Referring to the figure, microcode uses the processor interface 1 to load the control register 2 and mask register 3 to initialize and enable the trace array function. The control register 2 contains the following functions:

o Controls selection of trace mode

- Store data every cycle.
- Store data every control word.
- Store data every instruction.

o Controls selection of data inputs used to feed the trigger logic 4

o Selects what type of trigger is used

- 'OR' of trigger inputs.
- Compare trigger inputs with data in a portion of the mask register 3.
- External trigger 8.

o Enable one or more of the trigger types

o Selects the interrupt type issued to the processor

- Trap.
- Exception.

o Contains a constant for the number of events saved in the trace array after a trigger occurs.

- The mask register 3 contains the mask data and compare data used by the trigger logic 4. The mask data is used to ignore (don't care) specified bits in the trigger logic data paths. The compare data is used to form the compare trigger.
- The trace signals 7, stored in the array 6, includes major control interfaces and sequencers. These signals may or may not be available external to the multi-chip module.
- Once the trace array is set up, data is continually stored into the array 6 until a trigger occurs. The data is stored in a circular fashion with the first in being the the first replaced once the array address has wrapped around back to address zero. Addressing the array 6 is handled by the array control 5 logic. Based on data in the control register 2, the address will be incremented every cycle, every control word, or every instruction with data being stored to that address.
- Once a trigger occurs, a processor interrupt 9

is issued. The interrupt may be an exception or a trap. The processor will handle a trap immediately while waiting to handle an exception until the end of an instruction.

- After the microcode completes handling the processor interrupt 9, it will reinitialize the trace array function using the processor interface 1. The trace array is then ready for the next trigger.
- The external trigger 8 is shared by all the trace arrays in a multiple processor system. It allows the trigger in one processor to stop traces in all the other processors' trace arrays, thus synchronizing the traces in all the processors. The external trigger 8 signal can also be used to trigger the on module arrays with an external signal (from an external logic analyzer or other external logic) or to trigger an external logic analyzer (see the figure).

TDB-ACC-NO: NA9306335

DISCLOSURE TITLE: Recovery Procedure for Data Channel

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, June 1993, US

VOLUME NUMBER: 36

ISSUE NUMBER: 6A

PAGE NUMBER: 335 - 336

PUBLICATION-DATE: June 1, 1993 (19930601)

CROSS REFERENCE: 0018-8689-36-6A-335

DISCLOSURE TEXT:

An I/O (input/output) subsystem in a data processing system is made up of I/O devices and a system of busses and processors that transfer data between the devices and central processor memory. A magnetic disk storage device is a familiar example of an I/O device.

In some I/O subsystems, processors called the channel processors (or channels) handle data transfers, and a processor called the IOP communicates with the operating system in the central processor to begin an operation and later to report that the transfer has completed successfully or that it has failed. A processor called the process controller (PC) handles other situations such as errors.

When an error is detected by the channel, the channel reports separately to the IOP and the PC, and the IOP reports to the operating system.

- This channel recovery procedure begins when a latch is set

signifying that an error has occurred on the channel bus (an interface check) or in the channel processor (a channel check). When a channel check occurs, the clocks in the channel are stopped and normal channel operations end. When an interface check occurs, for example, a parity error on the bus from the device, the clocks in the channel are not stopped, and normal channel operation can continue with other devices on the bus. The general recovery procedure has the following steps: (1) the channel signals the device to disconnect from the channel;

(2) the channel puts data into a channel memory called the trace array and sends an interrupt to the PC and the PC reads the trace array and begins its part of the recovery procedure;

(3) the channel reports to the IOP and the IOP reports to the operating channel; and (4) the operating system issues instructions to the channel (Halt, Clear Subchannel) to retry or restart the operation.

- Sometimes a second error occurs before the PC has read the trace data for the first error. In this recovery procedure, the channel checks the status of the first trace data and it does not enter the trace data into the trace array in this situation. The channel proceeds with the rest of the recovery procedure, and the data that would have been entered in the trace array is lost.

- A normal data transfer proceeds through a series of stages, and the channel keeps track of the stage it is in. As part of this recovery procedure, the channel identifies this stage to the operating system. The operating system uses this

information to decide whether a simple retry is possible or whether the operation must be restarted from the beginning. As an example, if an operation of reading or writing a tape has proceeded to the stage where the tape has been moved past the read head, the operation can not be retried but must be restarted.

- At an early stage of a data transfer, the channel gets control of the subchannel information block for the device that is to take part in the data transfer. If the subchannel information block is not available, the channel continues to try to get control of it. If an error occurs during this time, the operating system follows a different procedure than for errors that occur after the channel has acquired the subchannel information block. As part of the channel recovery procedure, the channel reports this information to the operating system.
- In a normal data transfer, the controller for the device raises a tag line to transfer a byte in either direction and the channel raises a corresponding tag to accompany a byte from the channel or to acknowledge a byte from the device. Some channels report a residual byte count, which is zero if the numbers or tags are equal. When a channel operating in this mode executes this recovery procedure, it calculates a residual byte count and sends it to the operating system. This count is not an exact count as in a normal data transfer, but it is within a range that permits the operating system to identify generally where the error occurred.
- When this recovery procedure has been completed, the channel

goes into an idle state waiting for the next operation it is to perform.

DOCUMENT-IDENTIFIER: US 20020186507 A1

TITLE: Conductor assembly for reducing track misregistration in a disk drive

---------- KWIC ---------

Summary of Invention Paragraph - BSTX (4):

[0002] Disk drives are widely used in computers and data processing systems for storing information in digital form. These disk drives commonly use one or more rotating storage disks to store data in digital form. Each storage disk typically includes a data storage surface on each side of the storage disk. These storage surfaces are divided into a plurality of narrow, annular regions of different radii, commonly referred to as "tracks". Typically, a head stack assembly having a positioner and an E-block including an actuator hub is used to position a data transducer of a transducer assembly proximate each storage surface of each storage disk. The data transducer transfers information to and from the storage disk when precisely positioned on the appropriate track (also known as a "target track") of the storage surface. A conductor assembly, including one or more trace arrays, electrically connects each data transducer to a drive circuitry.

Summary of Invention Paragraph - BSTX (7):

[0005] Prior art FIGS. 1A and 1B illustrate a portion of a prior art disk drive. More specifically, FIGS. 1A and 1B illustrate a conventional actuator arm 22P, a conductor assembly 32P including portions of two trace arrays 36P, and a transducer assembly 28P (shown only in FIG. 1A) that

are secured to the actuator arm 22P. The trace arrays 36P are generally flexible structures that run from the data transducer 70P, along the actuator arm 22P (only a portion is shown in FIG. 1A), to the drive circuitry (not shown in FIGS. 1A and 1B). Each trace array 36P typically includes a flexible, middle span 72BP that bows away from the actuator arm 22P.

Summary of Invention Paragraph - BSTX (8):

[0006] One of the major drawbacks of conventional flexible trace arrays is that the turbulent airflow in the disk drive causes the trace arrays 36P to be intermittently driven into resonance. This motion of the conductor assembly 32P can pull the data transducer 70P off-track, creating errors known as track misregistration. Specifically, the non-repeatable component of track misregistration, known as "non-repeatable runout" (NRRO) is particularly impacted by the air turbulence created by the storage disks. In fact, the extent of the track misregistration increases exponentially with higher storage disk rotation rates.

Detail Description Paragraph - DETX (2):

[0031] Referring initially to FIG. 2, a disk drive 10 according to the present invention includes a drive housing 12, a disk assembly 14 including one or more storage disks 16, and a head stack assembly 18 including (i) an E-block 20 with one or more actuator arms 22 that each have a first surface 23 lying in a first plane 24 and a second surface 25 lying in a second plane 26 (not shown on FIG. 2), (ii) one or more transducer assemblies 28 secured to each actuator arm 22, (iii) a positioner 30, and (iv) a conductor assembly 32 for conveying electrical signals between the transducer assembly 28 and a

drive circuitry 34. The conductor assembly 28 includes one or more trace arrays 36, each having a trace segment 38 that is generally situated along the actuator arm 22, as illustrated in the embodiments shown in FIGS. 3A, 5A, 6 and 7A.

Detail Description Paragraph - DETX (16):

[0045] The conductor assembly 32 structurally and electrically connects each data transducer 70 to the drive circuitry 34. The design of the conductor assembly 32 can vary depending upon the requirements of the head stack assembly 18 and the disk drive 10. For example, FIGS. 3A, 4, 5A-5B, 6 and 7A illustrate five different embodiments. In each of these embodiments, the conductor assembly 32 includes at least a first trace segment 38A for each actuator arm 22 in the disk drive 10. Each trace segment 38 carries electrical signals along a portion of the trace array 36 between one of the transducer assemblies 28 and the drive circuitry 34. The materials which form the trace segments 38 can vary depending upon the requirements of the disk drive 10. The trace segments 38 can be formed from copper or other suitable metals, and can also include a substrate material such as polyimide or other appropriate plastics or metals, as examples.

Detail Description Paragraph - DETX (19):

[0048] A first embodiment of the conductor assembly 32 is illustrated in FIG. 3A. In this embodiment, the conductor assembly 32 includes at least the first trace segment 38A and a second trace segment 38B. Additional trace segments (not shown) can also be used with the present invention. The second trace segment 38B also includes the middle span 72B that is generally

unsupported by the actuator arm 22. In this embodiment, the design and quantity of the trace segments 38A, 38B, can vary depending upon the requirements of the disk drive 10. Each trace segment 38A, 38B, forms a portion of the trace array 36 (only partially shown in FIG. 3A). The trace array 36 preferably extends from one of the data transducers 70, substantially along the actuator arm 22, to the drive circuitry 34 of the disk drive 10.